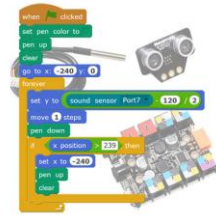


0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



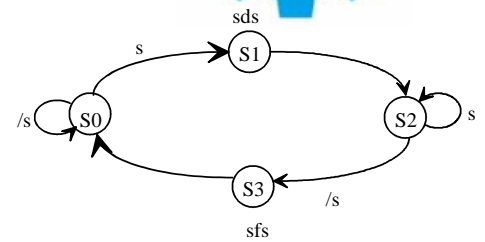
mBlock programmation avancée

Makeblock



1 Limite du suivi de ligne classique

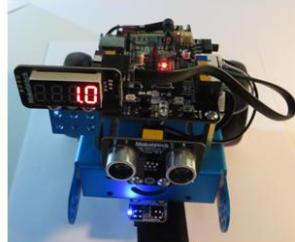
Le programme suiveur de ligne tel que présenté dans le document suivi de ligne présente quelques limitations. En effet il est impossible dans le cas où le robot a quitté la piste d'avoir un comportement différent selon que l'on était en virage droit ou gauche car dans ce cas le capteur de suivi de ligne donne la même information et le programme réagit toujours de la même manière :



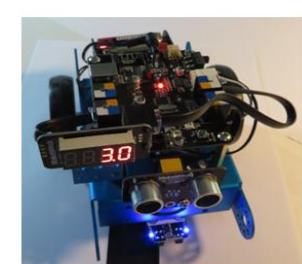
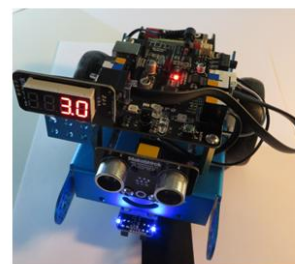
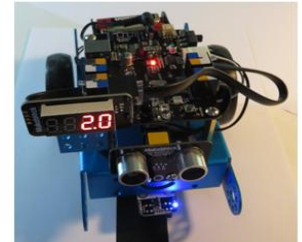
```

répéter indéfiniment
mettre Suivi Ligne à état du suiveur de ligne sur le Port2
sur le 7 segments du Port1 afficher Vitesse
si Suivi Ligne = 0 alors
    activer le moteur M1 à la puissance 150
    activer le moteur M2 à la puissance 150
sinon
    si Suivi Ligne = 1 alors
        tourner à gauche à la vitesse 40
    sinon
        si Suivi Ligne = 2 alors
            tourner à droite à la vitesse 40
        sinon
            reculer à la vitesse 50
    
```

Virage à gauche

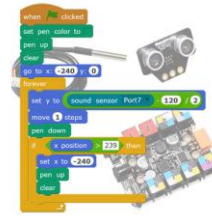


Virage à droite



Pour ces deux situations du robot nous avons la même information issue du capteur de suivi de ligne 3.0.

0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010



2 Un peu de théorie

2.1 Automate séquentiel (version light)

Un système pour lequel le passé n'intervient pas dans le calcul de l'état futur du système est de nature combinatoire. C'est le cas de notre premier programme, pour chaque situation du capteur de suivi de ligne correspond une situation réponse du robot, avancer, reculer, tourner à droite à gauche.

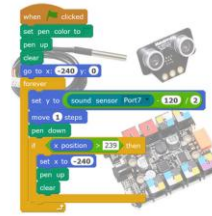
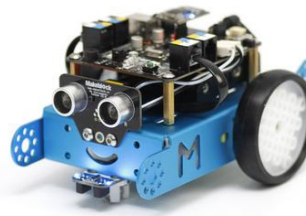
Si nous voulons améliorer la gestion du comportement du robot il faut faire intervenir la mémoire. En effet c'est en se souvenant de la situation précédente que l'on peut interpréter la prochaine action à réaliser, on ne réagira pas uniquement sur l'état du capteur d'entrée mais sur le profil de la piste en détectant un virage à droite ou à gauche. Il ne faut plus réagir en fonction de l'état du capteur de suivi de ligne mais interpréter cette information pour déduire la situation du robot et donc son comportement à venir.

Réagir en tenant compte du passé du système consiste à comprendre le fonctionnement du robot comme celui d'un automate séquentiel : la situation future dépend de la situation présente (capteurs d'entrées) et de l'histoire du système.

Le robot ne peut exécuter qu'une seule action à la fois, (pour les mouvements et avec la programmation mBlock), son comportement peut donc se traduire par un automate séquentiel de type machine de Moore.

Une machine de Moore est un automate dont les sorties sont associées aux états. Elles sont adaptées à la description de systèmes séquentiels dont le comportement peut être traduit par un chronogramme. Dans un chronogramme on ne peut être à deux endroits en même temps de ce fait il n'y a qu'un seul état actif à la fois. Illustrons ces principes sur un exemple simple.

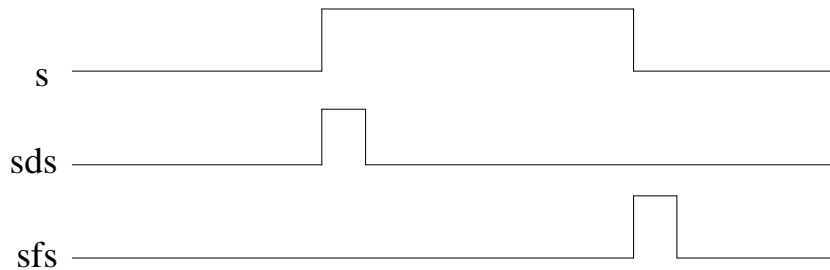
0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



2.2 Détection de début et de fin d'impulsion

Étudions le problème suivant très courant en électronique des signaux à savoir la détection automatique du début et de la fin d'une impulsion notée ici 's'.

Comportement attendu de notre système

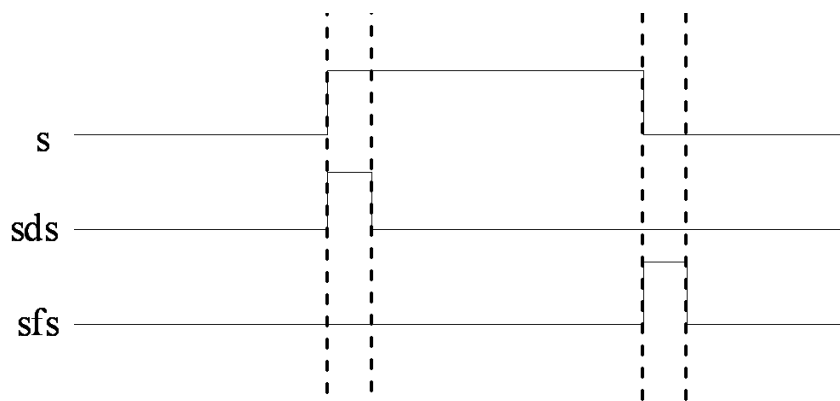


↳ Le système ici décrit par son chronogramme est-il séquentiel ?

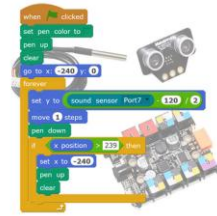
↳ Quelles sont les entrée(s) sortie(s) ?

↳ Décrire le comportement attendu du système par une courte phrase :

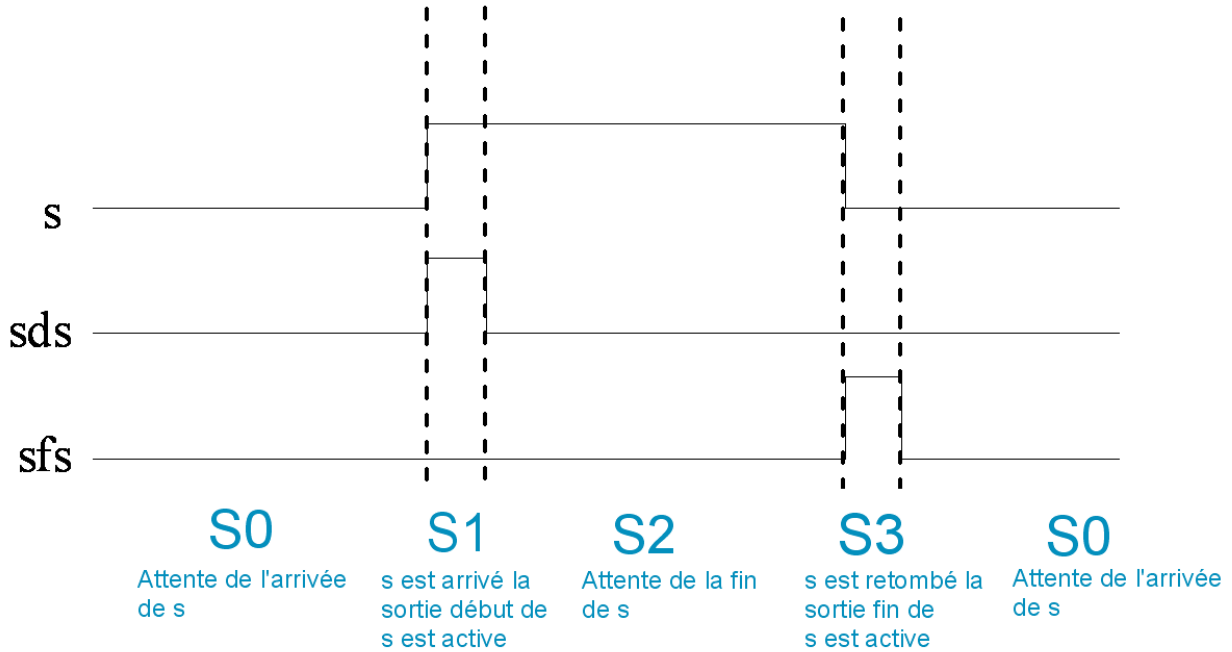
↳ Découpons notre chronogramme en tranche correspondant à toutes les situations rencontrées :



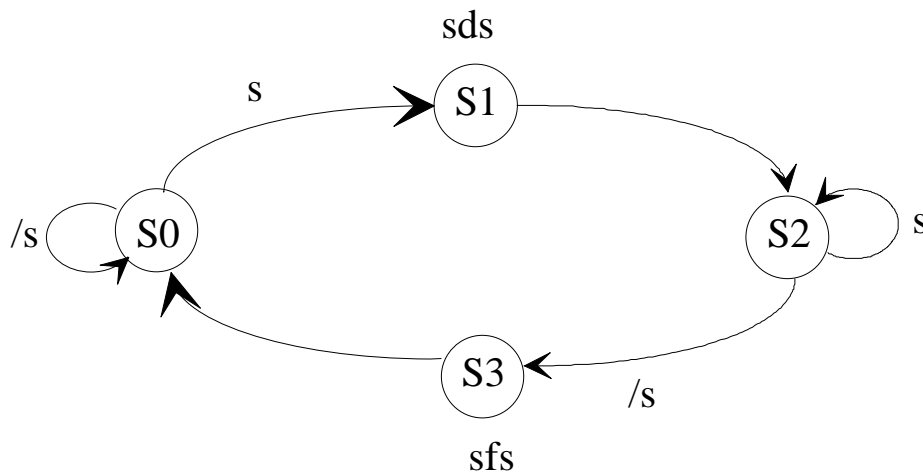
0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



↳ Nommons les tranches, chacune d'entre elles correspond à un état de notre système :



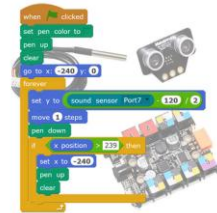
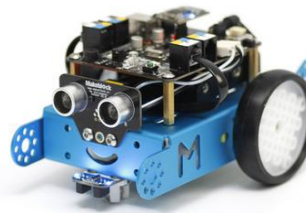
↳ Traduction en automate de Moore :



Nous observons que les états sont traduits par des cercles, les sorties associées sont indiquées à cotés du cercle. Les transitions entre états sont indiquées par des flèches avec la condition d'activation explicite (équation logique, état d'une variable) ou implicite (au prochain cycle).

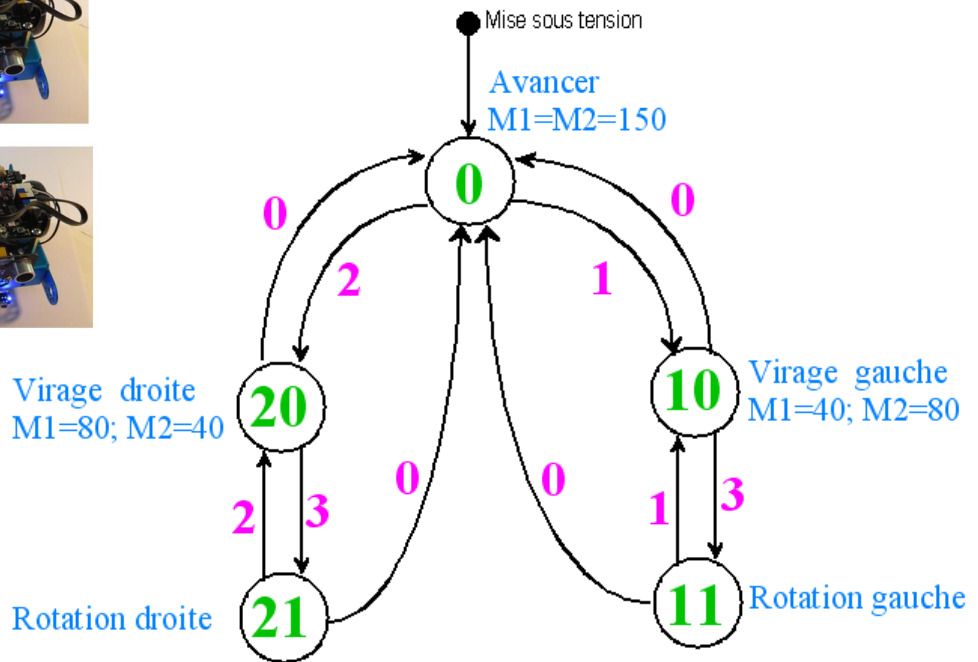
Un seul état est actif à la fois (ce n'est pas un grafcet, ni un réseau de Pétri).

0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010

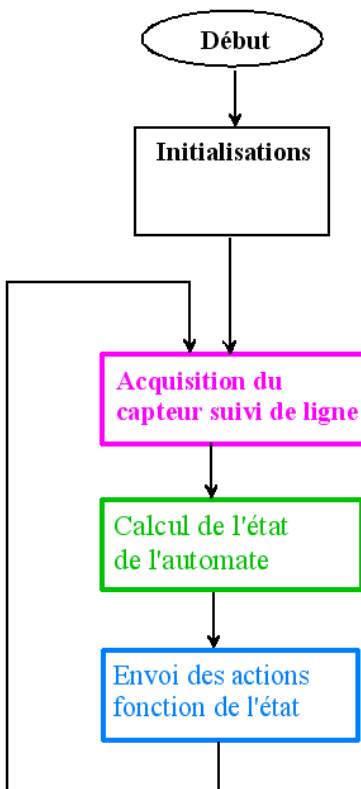
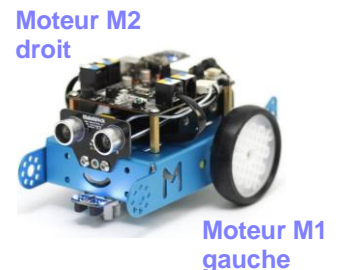


3 Mise en œuvre dans la commande du robot

Nous pouvons décrire le comportement attendu de notre robot par l'automate ci-dessous :



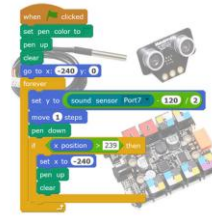
action =
 suivi de ligne =
 état de l'automate



L'organigramme de notre automate est donné ci-contre.

- ↪ L'état sera représenté par une variable nommée mode
- ↪ L'état futur par une variable nommée nouveau mode
- ↪ Les sorties seront calculées en fonction de cette variable mode et non plus en fonction directe du capteur de suivi de la ligne.
- ↪ C'est le capteur de suivi de ligne qui provoquera l'évolution de notre automate.

0100100100**1**0100100
 101001**1**01001000100
 0101**0**0010010010010



3.1 Implantation avec mBlock

```
mettre Mode à 0
mettre Nouveau Mode à 0
```

Initialisation

répéter indéfiniment

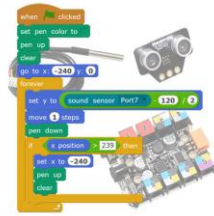
```
mettre Suivi Ligne à état du suiveur de ligne sur le Port2
sur le 7 segments du Port1 afficher Mode
```

Lecture des entrées

```
si Mode = 0 et Suivi Ligne = 1 alors
  mettre Nouveau Mode à 10
si Mode = 0 et Suivi Ligne = 2 alors
  mettre Nouveau Mode à 20
si Mode = 10 et Suivi Ligne = 3 alors
  mettre Nouveau Mode à 11
si Mode = 10 et Suivi Ligne = 0 alors
  mettre Nouveau Mode à 0
si Mode = 11 et Suivi Ligne = 0 alors
  mettre Nouveau Mode à 0
si Mode = 11 et Suivi Ligne = 1 alors
  mettre Nouveau Mode à 10
si Mode = 20 et Suivi Ligne = 3 alors
  mettre Nouveau Mode à 21
si Mode = 20 et Suivi Ligne = 0 alors
  mettre Nouveau Mode à 0
si Mode = 21 et Suivi Ligne = 0 alors
  mettre Nouveau Mode à 0
si Mode = 21 et Suivi Ligne = 2 alors
  mettre Nouveau Mode à 20
mettre Mode à Nouveau Mode
```

Calcul de l'évolution de l'état de l'automate

0100100100**1**0100100
101001**1**01001000100
0101**0**0010010010010



```
si Mode = 0 alors
  activer le moteur M1 à la puissance 151
  activer le moteur M2 à la puissance 150
si Mode = 10 alors
  activer le moteur M1 à la puissance 40
  activer le moteur M2 à la puissance 80
si Mode = 20 alors
  activer le moteur M1 à la puissance 80
  activer le moteur M2 à la puissance 40
si Mode = 11 alors
  tourner à gauche à la vitesse 50
si Mode = 21 alors
  tourner à droite à la vitesse 50
```

Envoi des actions